STUDY OF THE DESIGN AND TUNING METHODS OF PID CONTROLLER BASED ON FUZZY LOGIC AND GENETIC ALGORITHM

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor in Technology

in

Electronics and Instrumentation Engineering

by

Sangram Keshari Mallick

and

Mehetab Alam Khan

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela

May, 2011

STUDY OF THE DESIGN AND TUNING METHODS OF PID CONTROLLER BASED ON FUZZY LOGIC AND GENETIC ALGORITHM

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor in Technology

in

Electronics and Instrumentation Engineering

by

Sangram Keshari Mallick

and

Mehetab Alam Khan

under the guidance of

Prof. G. S. Rath

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela

May, 2011

CERTIFICATE

This is to certify that the thesis entitled "**Study of the design and tuning methods of PID controller based on fuzzy logic and genetic algorithm**" submitted by **Sangram Keshari Mallick** (107EI027) and **Mehetab Alam Khan** (107EI028) in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electronics and Instrumentation Engineering at National Institute of Technology, Rourkela is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in thesis has not been submitted to any other university/ Institute for the award of any degree or Diploma.

Place: Rourkela
Date:

Prof. G. S. Rath
Dept. of Electronics & Communication Engineering
National Institute of Technology
Rourkela – 769008

ABSTRACT

This project tries to explore the potential of using soft computing methodologies in controllers and their advantages over conventional methods. PID controller, being the most widely used controller in industrial applications, needs efficient methods to control the different parameters of the plant. This thesis asserts that the conventional approach of PID tuning is not very efficient due to the presence of non-linearity in the system of the plant. The output of the conventional PID system has a quite high overshoot and settling time.

The main focus of this project is to apply two specific soft-computing techniques viz. fuzzy logic and genetic algorithm to design and tuning of PID controller to get an output with better dynamic and static performance. The application of fuzzy logic to the PID controller imparts it the ability of tuning itself automatically in an on-line process while the application of genetic algorithm to the PID controller makes it give an optimum output by searching for the best set of solutions for the PID parameters.

The project also discusses the benefits and the short-comings of both the methods. The simulation outputs are the MATLAB results obtained for a step input to a third-order plant.

ACKNOWLEDGEMENT

We would like to express our gratitude and sincere thanks to Prof. G. S. Rath whose guidance, support and continuous encouragement helped us being motivated towards excellence throughout the course of this work. We wish to extend our gratitude to Prof. T. K. Dan for his timely suggestions and guidance during the project work.

Finally, we would like to thank all of them who have been helpful and were associated with us directly and indirectly throughout the course of our work.

Sangram Keshari Mallick
107EI027

Mehetab Alam Khan
107EI028

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Conventional PID controllers

1.1.1 Preliminary and background

PID controllers are the most widely-used type of controller for industrial applications. They are structurally simple and exhibit robust performance over a wide range of operating conditions. In the absence of the complete knowledge of the process these types of controllers are the most efficient of choices. The three main parameters involved are Proportional (P), Integral (I) and Derivative (D). The proportional part is responsible for following the desired set-point, while the integral and derivative part account for the accumulation of past errors and the rate of change of error in the process respectively.

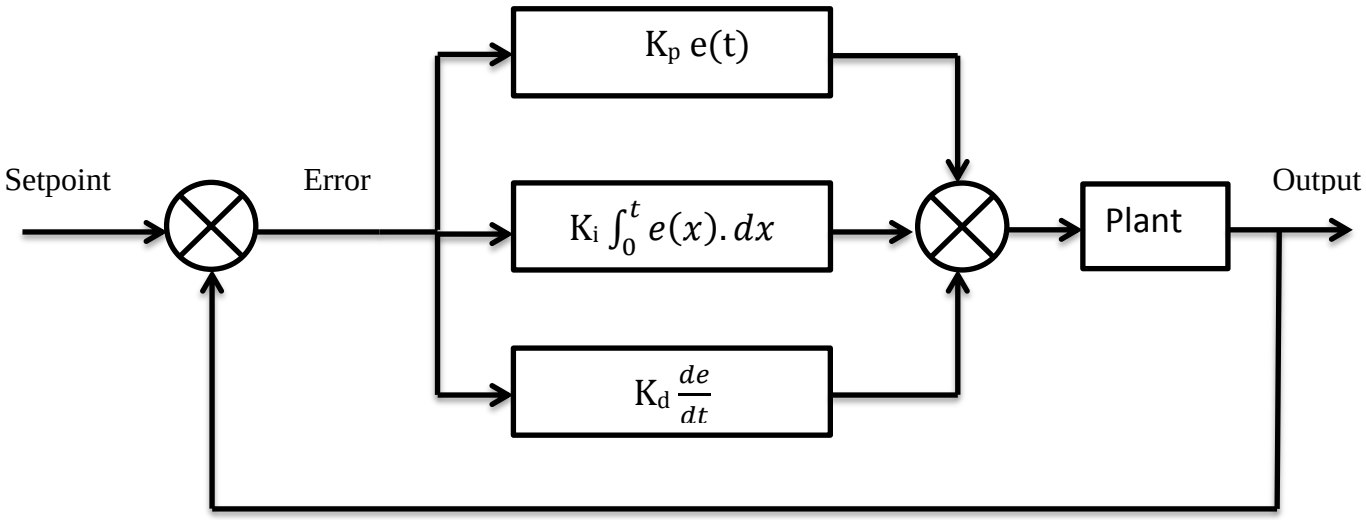


Fig. 1.1 Basic block diagram of a conventional PID controller

For the PID controller presented in Fig. 1.1,

$$\text{Output of the PID controller, } u(t) = K_p\, e(t) + K_i \int_0^t e(x).dx + K_d \frac{de(t)}{dt} \qquad \ldots\ldots 1.1$$

Where,

Error, e(t) =Setpoint- Plant output

K_p = proportional gain, K_i = integral gain, K_d = derivative gain

1.1.2 Tuning of PID parameters

Tuning of a PID controller refers to the tuning of its various parameters (P, I and D) to achieve an optimized value of the desired response. The basic requirements of the output will be the stability, desired rise time, peak time and overshoot. Different processes have different requirements of these parameters which can be achieved by meaningful tuning of the PID parameters. If the system can be taken offline, the tuning method involves analysis of the step-input response of the system to obtain different PID parameters. But in most of the industrial applications, the system must be online and tuning is achieved manually which requires very experienced personnel and there is always uncertainty due to human error. Another method of tuning can be Ziegler-Nichols method[8]. While this method is good for online calculations, it involves some trial-and-error which is not very desirable.

The transfer function describing the plant for our example is as follows:

$$G(s) = \frac{500}{s^3+30s^2+1000s} \qquad \ldots\ldots 1.2$$

The above transfer function will be used for the study and comparison of the output response of conventional PID, fuzzy PID and PID using genetic algorithm.

Following is an example of a conventional PID controller and its output to a step input response as achieved with some particular control parameter values. The output is the simulation result obtained with the help of MATLAB.

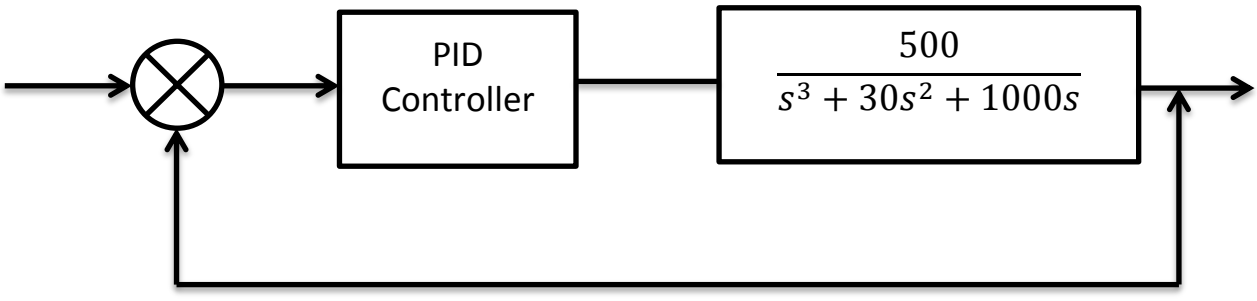
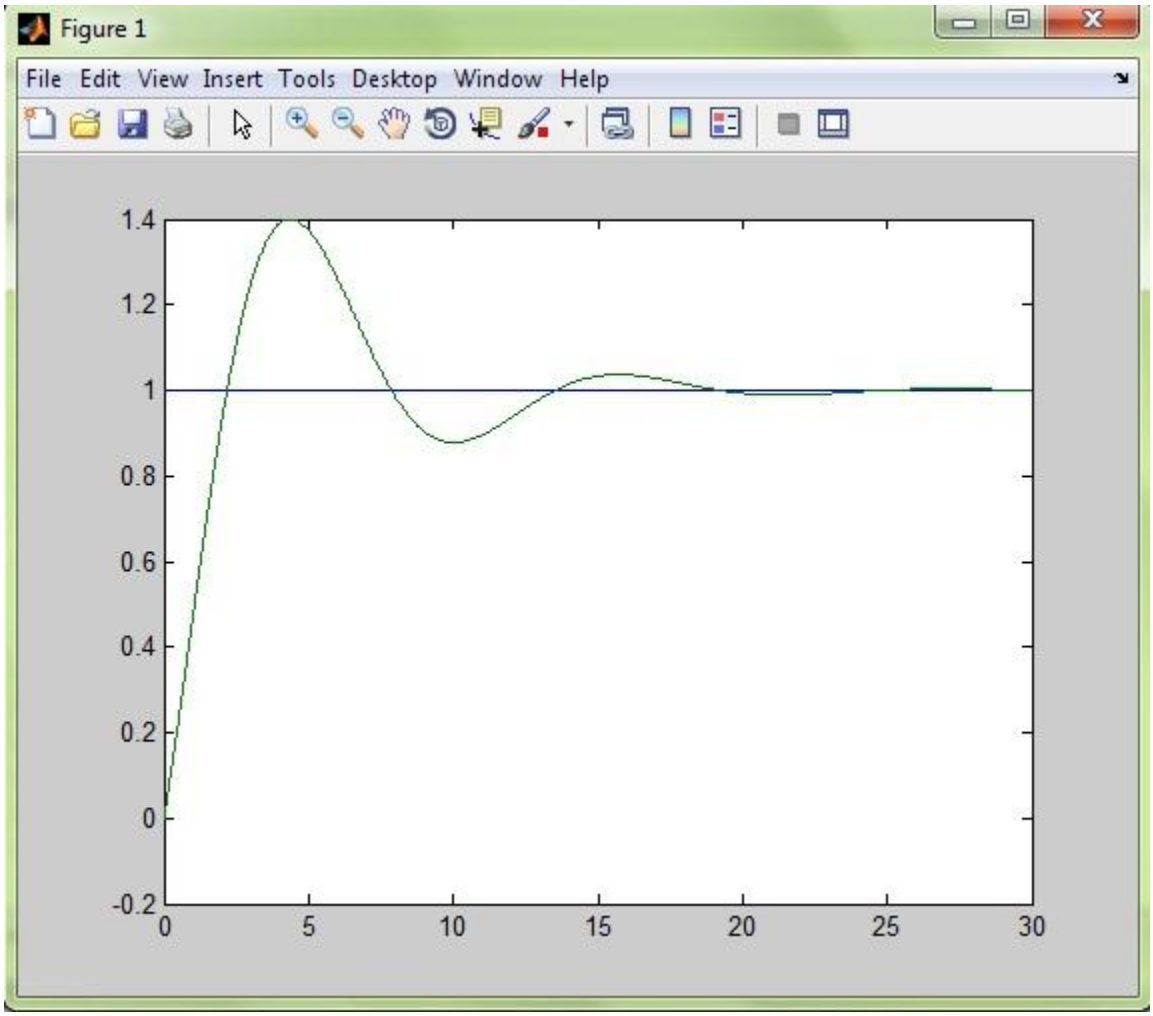


Fig. 1.2 Block diagram of the example system



$K_p = 0.6$ $K_i=0.5$ $K_d = 0.001$

Fig. 1.3 MATLAB simulation result for a conventional PID controller

Here, the output response of the given third order system has an overshoot of more than 40%. Settling time is also close to 20 seconds. For practical applications in industry these values are too high to be tolerated. An output response having a minimum overshoot and fastest response is required.

1.2 PID controller tuning using various soft-computing techniques

PID controller model structure needs to be very precise. But in practical applications, to a different extent, most of the industrial processes exist to be nonlinear, the variability of parameters and the uncertainty of model are very high, thus using conventional PID control the precise control of the process cannot be achieved.The common methods known for tuning require the process model to be of a certain type, for example as in the case of a 'First order plus dead time' model. These methods require the process model to be reduced if it's too complicated originally. The above problems can be well addressed by the application of soft-computing methods for tuning of the PID controller. These are specially useful for solving problems of computationally complicated and mathematically intraceable. This is due to the convenience of combining natural systems with intelligent machines effectively with the help of soft-computing methods. Among all these soft-computing methods available Neural network ,fuzzy logic and genetic algorithm are the most important ones.

Fuzzy logic mainly employ the mechanisms that are based on verbal power. Due to this fact it is responsible for dealing with the uncertainties present in the system.

Genetic algorithm has its roots originated from the genetic science which is a biological phenomenon. This method is useful for the process of random search from a huge pool of data.

By the implementation of the knowledge of fuzzy logic and genetic algorithm in PID controller, the system response of the plant cant be improved.The overshoot and the rise time of the response can be decreased and the dynamic performance of the system can also be improved.

1.3 Thesis organization

In this thesis, the first Chapter 1 discusses the characteristics of conventional controller and their shortcomings with the help of simulation results obtained using MATLAB for a third-order plant. Chapter 2 discusses about fuzzy self-tuning PID controller, their advantages over conventional methods and their shortcomings. GA PID controller is studied in Chapter 3 which also discusses its various advantages and disadvantages. Finally Chapter 4 concludes the thesis by shedding some light on the future scope of this work.

CHAPTER 2

FUZZY SELF-TUNING PID CONTROLLER

2.1Fuzzy logic background

Fuzzy logic is a logic having many values. Unlike the binary logic system, here the reasoning is not crisp ,rather it is approximate and having a vague boundary. The variables in fuzzy logic system may have any value in between 0 and 1 and hence this type of logic system is able to address the values of the variables those lies between completely truth and completely false. The variables are called lingusistic variables and each linguistic variable is described by a membership function which has a certain degree of membership at a particualr instance.

System based on fuzzy logic carries out the process of decision making by incorporation of human knowledge into the system. Fuzzy inference system is the major unit of a fuzzy logic system. The decision making is an important part of the entire system. The fuzzy inference system formulates suitable rules and based on these rules the decisions are made. This whole process of decision making is mainly the combination of concepts of fuzzy set thoery, fuzzy IF-THEN rules and fuzzy reasoning. The fuzzy inference system makes use of the IF-THEN statements and with the help of connectors present (such as OR and AND), necessary decision rules are constructed.

The basic Fuzzy inference system may take fuzzy inputs or crisp inputs depending upon the process and its outputs, in most of the cases, are fuzzy sets.

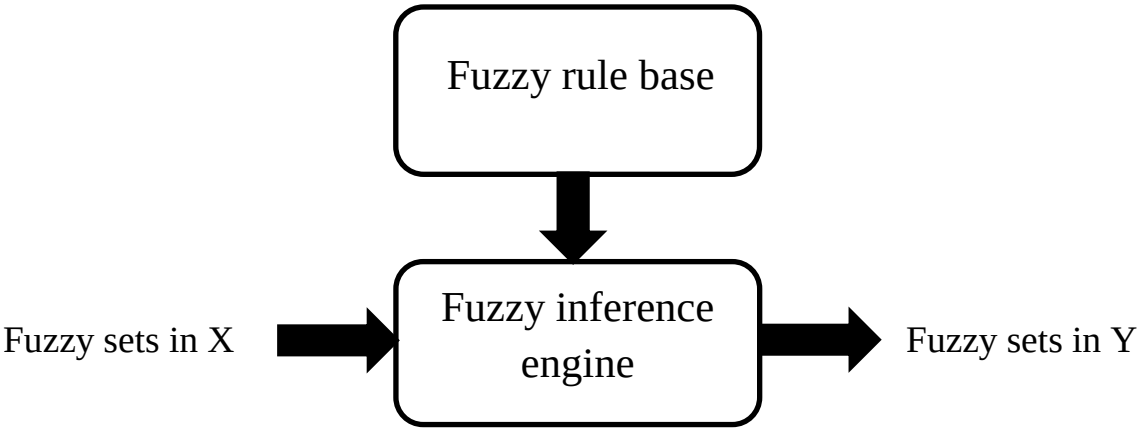


Fig. 2.1 A pure fuzzy system

The fuzzy inference system in Fig. 2.1 can be called as a pure fuzzy system due to the fact that it takes fuzzy sets as input and produces output that are fuzzy sets. The fuzzy rule base is the part responsible for storing all the rules of the system and hence it can also be called as the knowledge base of the fuzzy system. Fuzzy inference system is responisble for necessary decision making for producing a required output.

In most of the practical applications where the system is used as a controller, it is desired to have crisp values of the output rather than fuzzy set values. Therefore a method of defuzzification is required in such cases which converts the fuzzy values into corresponding crisp values.

In general there are three main types of fuzzy infernece systems such as :- Mamdani model, Sugeno model and Tsukamoto model. Out of these three, Mamdani model is the most popular one. There are also various defuzzification techniques such as :- Mean of maximum method, Centroid of area method, Bisector of area methd etc.

In this work Mamdani fuzzification technique[1] is used. There are two types of Mamdani fuzzy inference system such as, “min and max” and “product and max”. In our example, the “min and max” Mamdani system is used. For this type of system, min and max operators are used for AND and OR methods respectively. Fig. 2.2 explains the min and max Mamdani fuzzification technique. μ is the membership value for the linguistic varibales A1, B1, A2, B2, C1, C2 and C’. The fuzzy rules for the system are as follows:

1. If x is A1 and y belongs to B1, then z is C1.
2. If x is A2 and y belongs to B2, then z is to C2.

The defuzzification technique used is mean of maximum (MOM) method. As its name suggests it takes the mean value of z for which μ has maximum value.

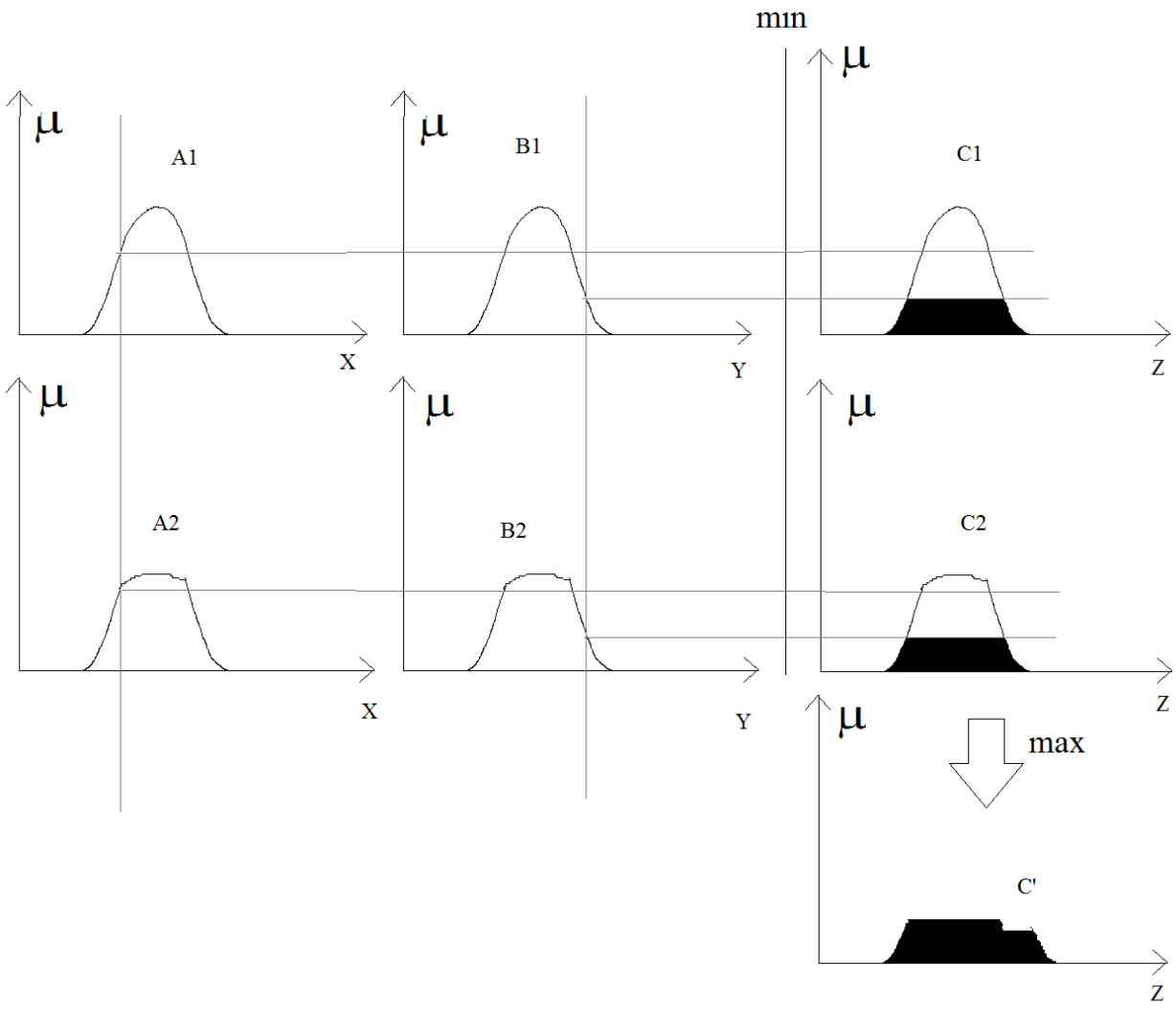


Fig. 2.2 min and max Mamdani fuzzy inference system

2.2 A self-tuning fuzzy PID controller

2.2.1 Self-tuning principle of fuzzy PID controller

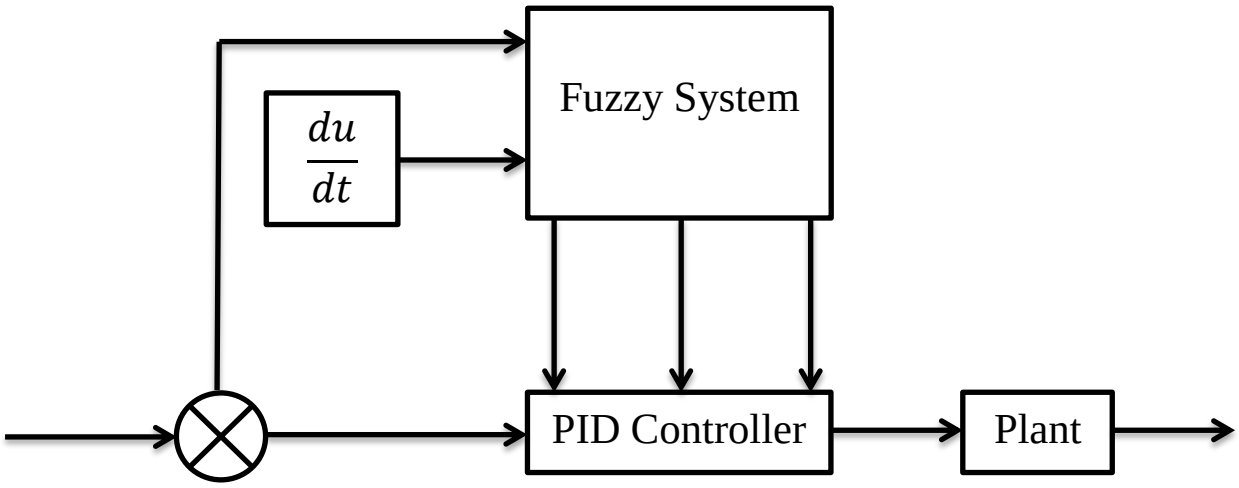
Fuzzy logic has been useful in recent years to formalize the ad-hoc approach of PID control. A fuzzy PID controller takes the conventional PID controller as the foundation which uses the fuzzy reasoning and variable universe of discourse to regulate the PID parameters. The chracteristics of a fuzzy system such as robustness and adaptabilty can be successfully incorporated into the controlling method for better tuning of PID parameters.

The term self-tuning refers to the characteristics of the controller to tune its controlling parameters on-line automatically so as to have the most suitable values of those parameters which result in optimization of the process output. Fuzzy self-tuning PID controller works on the control rules designed on the basis of theoretical and experience analysis. Therefore,it can tune the parameters K_p, K_i, and K_d by adjusting the other controlling parameters and factors on-line. This, in result makes the precision of overall contol higher and hence gives a better performance than the conventioanl PID controller or a simple fuzzy PID controller without self-tuning ability.

2.2.2 Design and structure of the self-tuning fuzzy PID controller

The Self-tuning fuzzy PID controller, which takes error "e" and rate of change-in-error "ec" as the input to the controller makes use of the fuzzy control rules to modify PID parameters on-line. The self-tuning of the PID controller refers to finding the fuzzy relationship between the three parameters of PID, K_p, K_i, and K_d and "e" and "ec", and according to the principle of fuzzy control modifying the three parameters in order to meet different requirements for control parameters when "e" and "ec" are different and making the control object produce a good dynamic and static performance. Selecting the language variables of “e” ,”ec”, K_p, K_i, and K_d is choosing seven fuzzy vlaues (NB, NM, NS, ZO, PS, PM, PB). The region of these varibales, in this case, is taken to be {-3,-2,-1,0,1,2,3}. Here (NB, NM, NS, ZO, PS, PM, PB) is the set of linguistic values which respectively represent “negative big”, ”negative medium”, ”negative small”, ”zero”, ”positive small”, ”positive medium” and “positive big”.

The following figure is the block diagram of a fuzzy self-tuning PID controller. As it can be seen from the block diagram, the fuzzification takes two inputs (e and ec) and gives three outputs ($\triangle K_p$, $\triangle K_i$, $\triangle K_d$).



Error, e =r(t)-c(t) Rate of change of error, ec=de/dt

Fig. 2.3 Basic structure of a fuzzy PID controller

The set of linguistic rules is the essential part of a fuzzy controller. In many cases it's easy to translate an expert's experience into these rules and any number of such rules can be created to define the actions of the controller. In the designed fuzzy system, conventional fuzzy conditions and relations such as :- "If e is A and ec is B,then K_p is C, K_i is D and K_d is E." are used to create the fuzzy rule table[3].

Table 2.1 Fuzzy rules for $\triangle K_p$

e \ ec	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

Table 2.2 Fuzzy rules for$\triangle K_i$

e \ ec	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

Table 2.3 Fuzzy rules for$\triangle K_d$

e \ ec	NB	NM	NS	ZO	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	NS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

Adaptive corrections can be made by the following methods,

$$K_p = K_p' + \triangle K_p$$

$$K_i = K_i' + \triangle K_i$$

$$K_d = K_d' + \triangle K_d$$

Here K_p', K_i', and K_d' refer to the previous value of the PID parameters whereas K_p, K_i, and K_d refer to the new corrected values of the parameters after a particular tuning step was completed.

2.2.3 Simulation results

The rsponse of the fuzzy self-tuning PID controller is obtained using Matlab. A two-input and three-output fuzzy controller is created and the membership functions and fuzzy rules are determined.

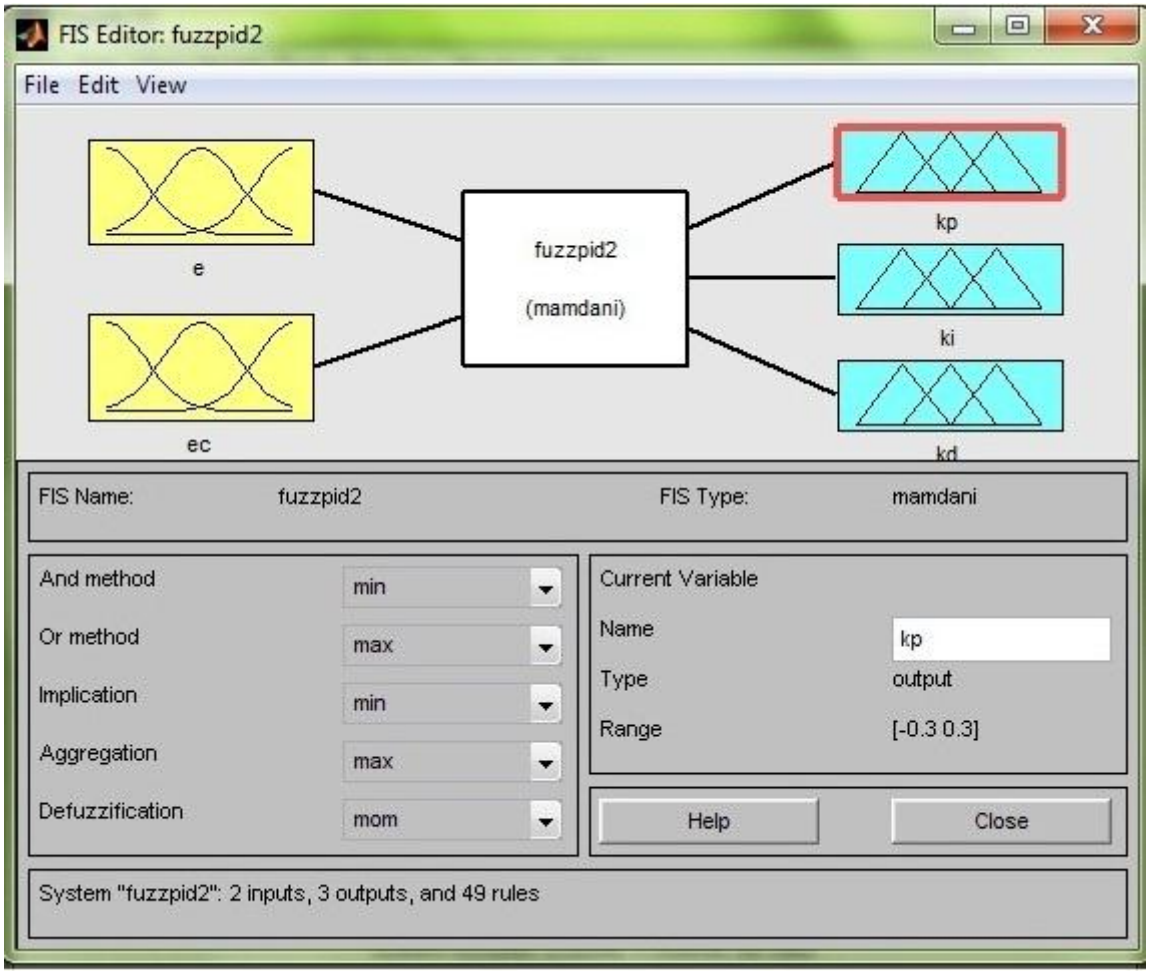


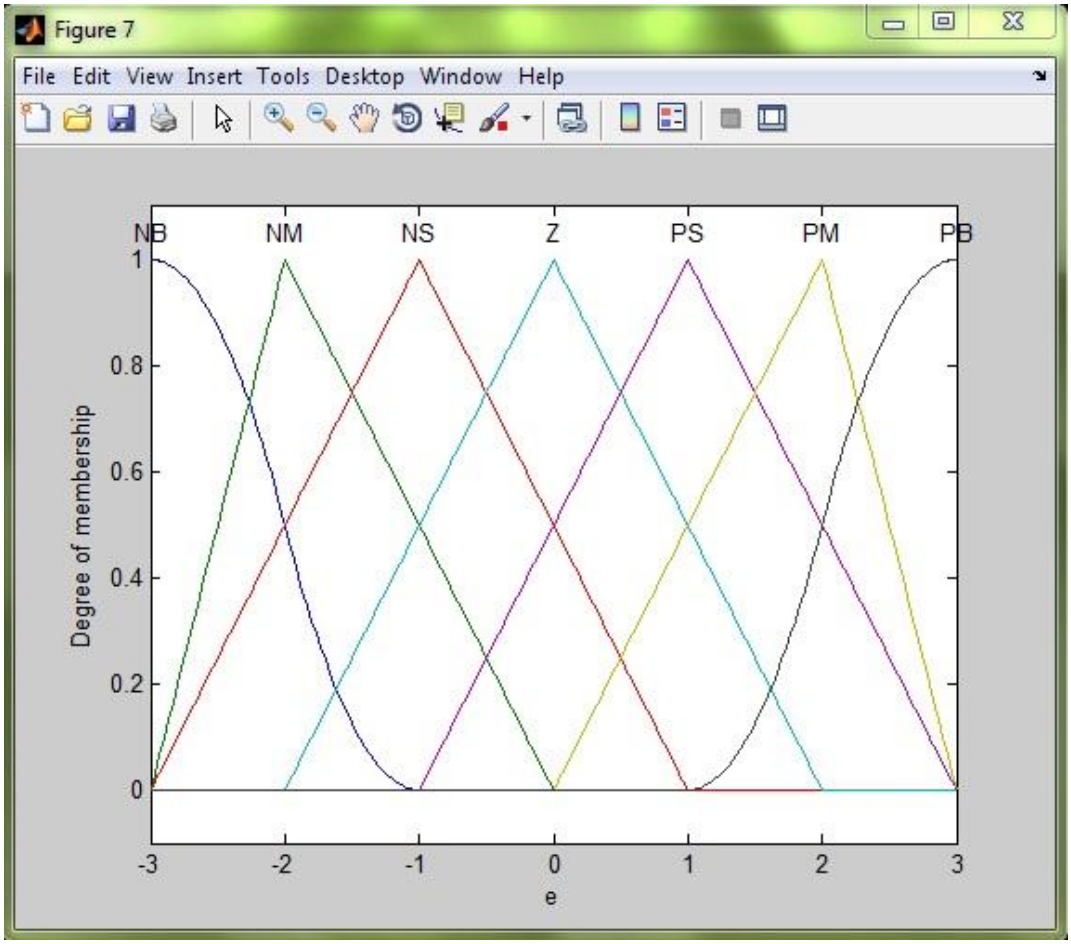
Fig. 2.4 Mamdani fuzzy system

The controller object is taken to be the third-order transfer fucntion,

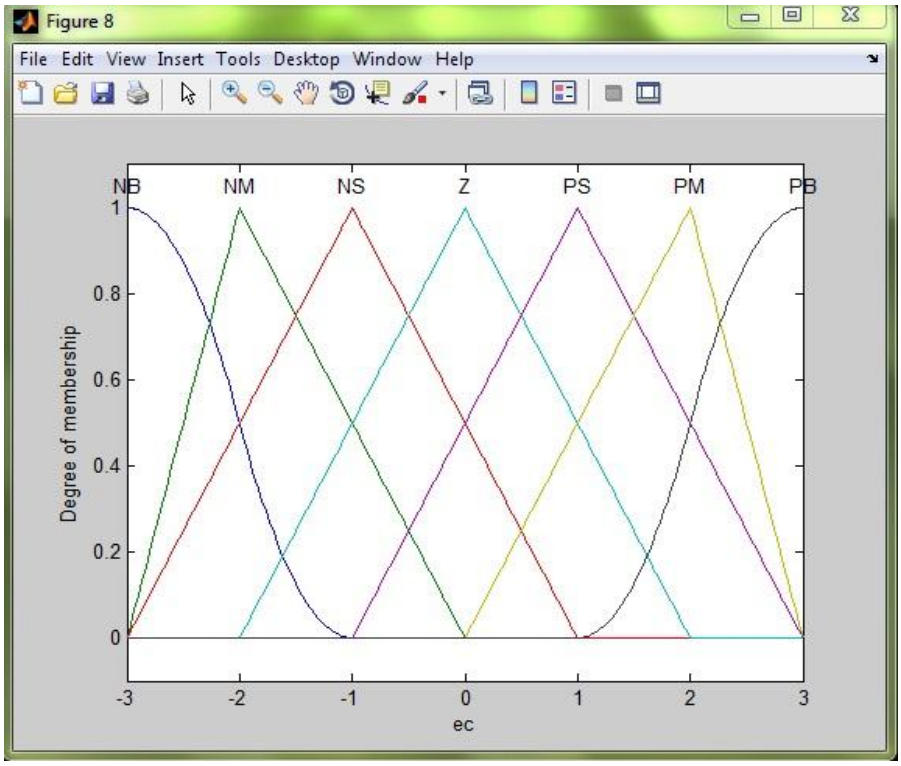
$$G(s) = \frac{500}{s^3+30s^2+1000s}$$

Debugging and setting the three parametrs K_p, K_i and K_d of the PID controller,their values are as follows : $K_p=0.6$ $K_i=0.5$ $K_d=0.001$

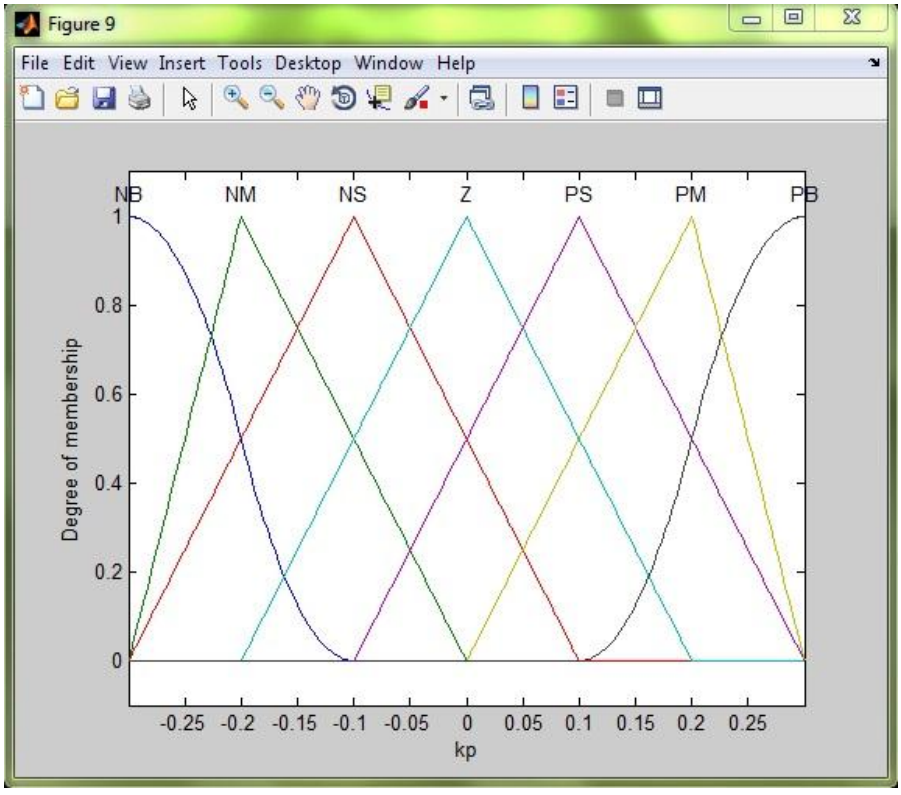
The membership function of the language variables “e”, “ec”, K_p, K_i and K_d are in the given range {-3,3} and their plots are as follows:



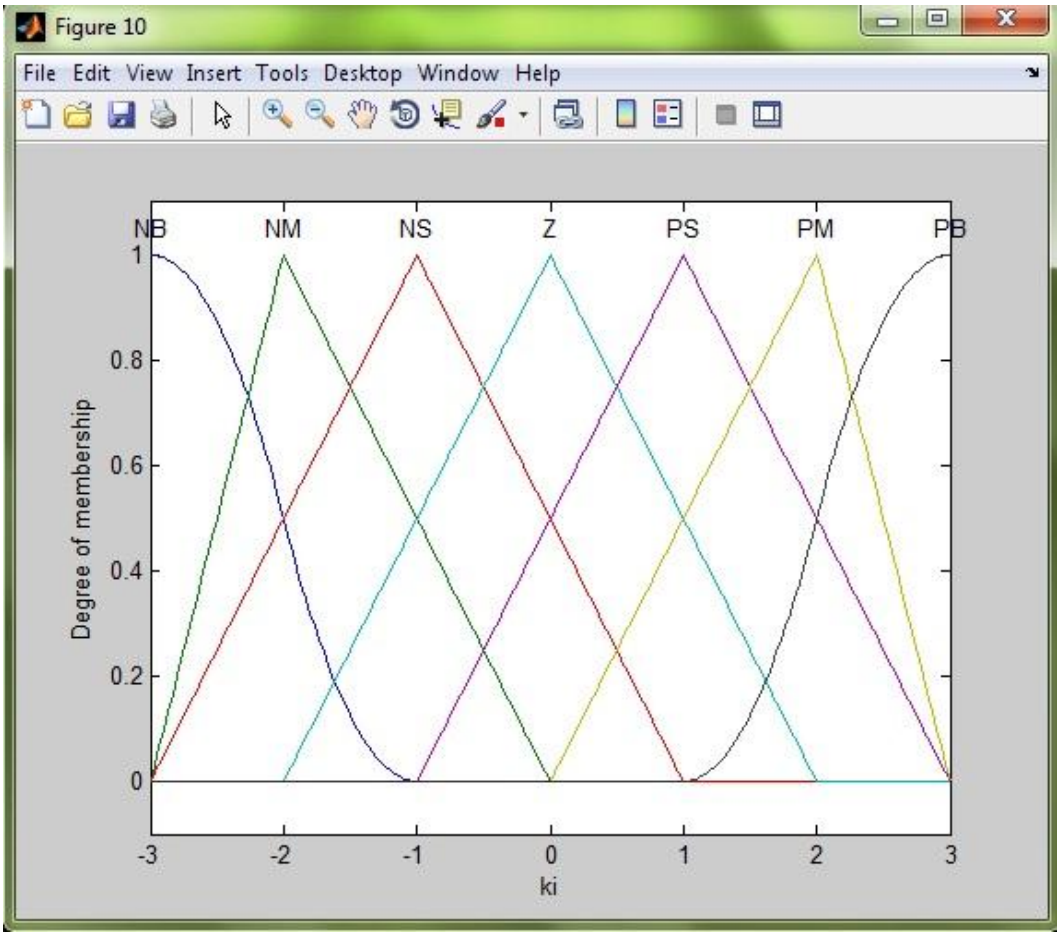
(a) Membership function of e



(b) Membership function of ec



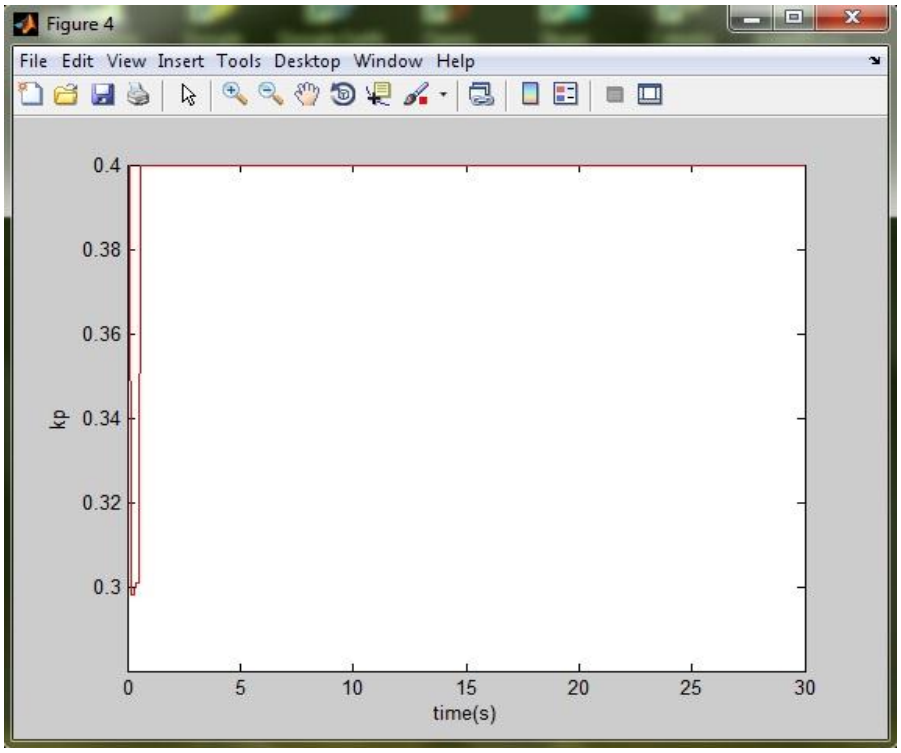
(c) Membership function of K_p



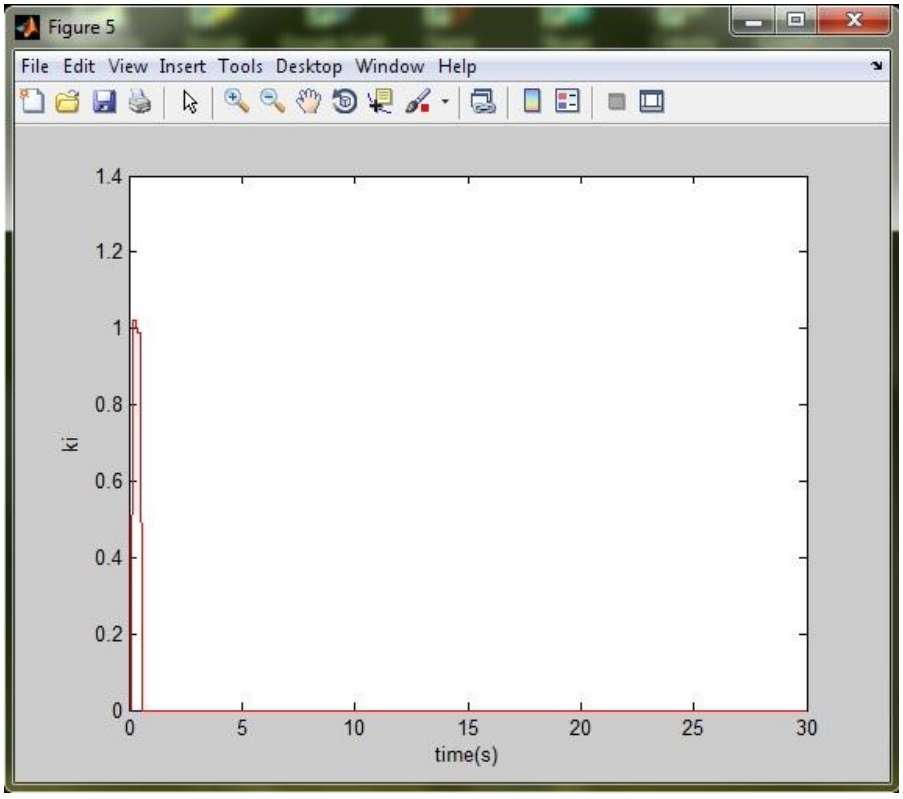
(d) Membership function of K_i

Fig. 2.5 Membership functions plots

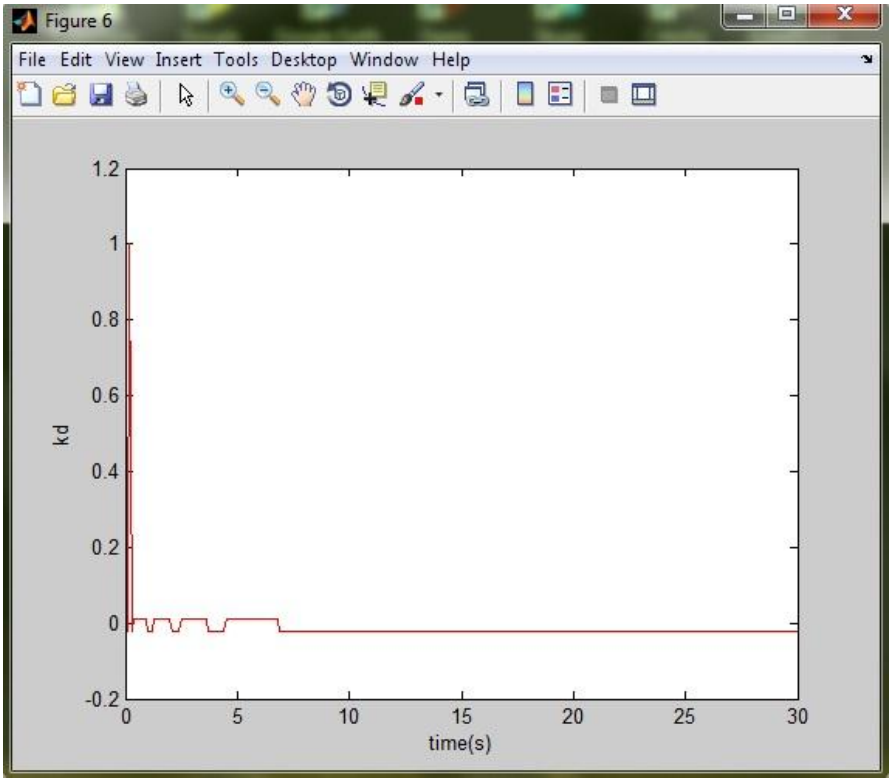
The output is the response of the system to a step input.The simluation results as obtained by Matlab are given in Fig. 2.6.



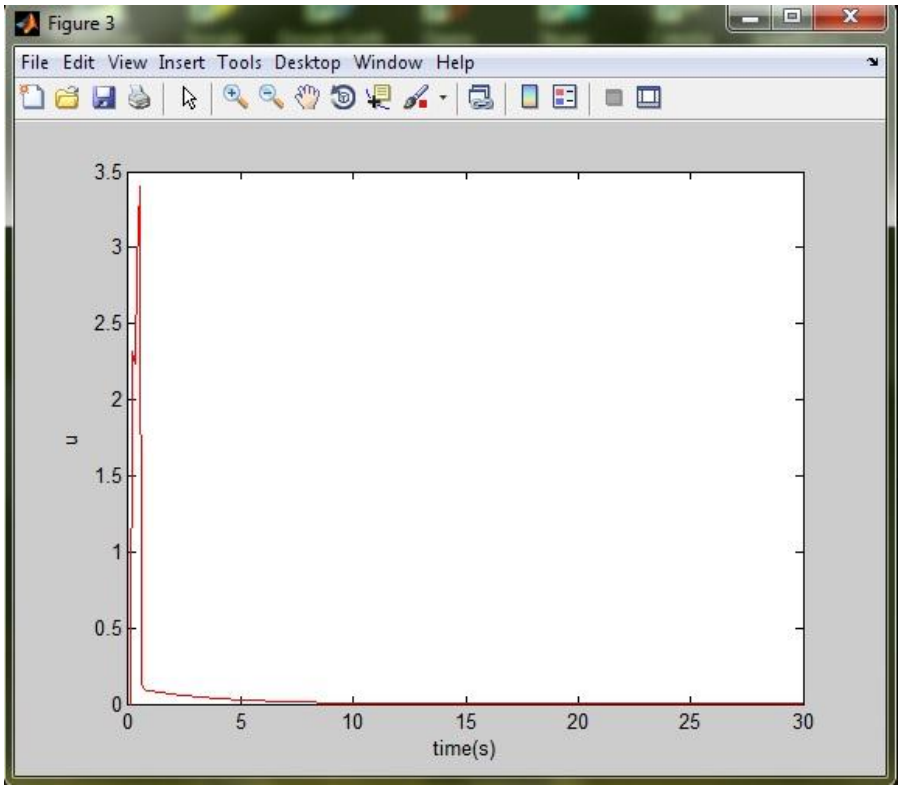
(a) Optimized value of K_p



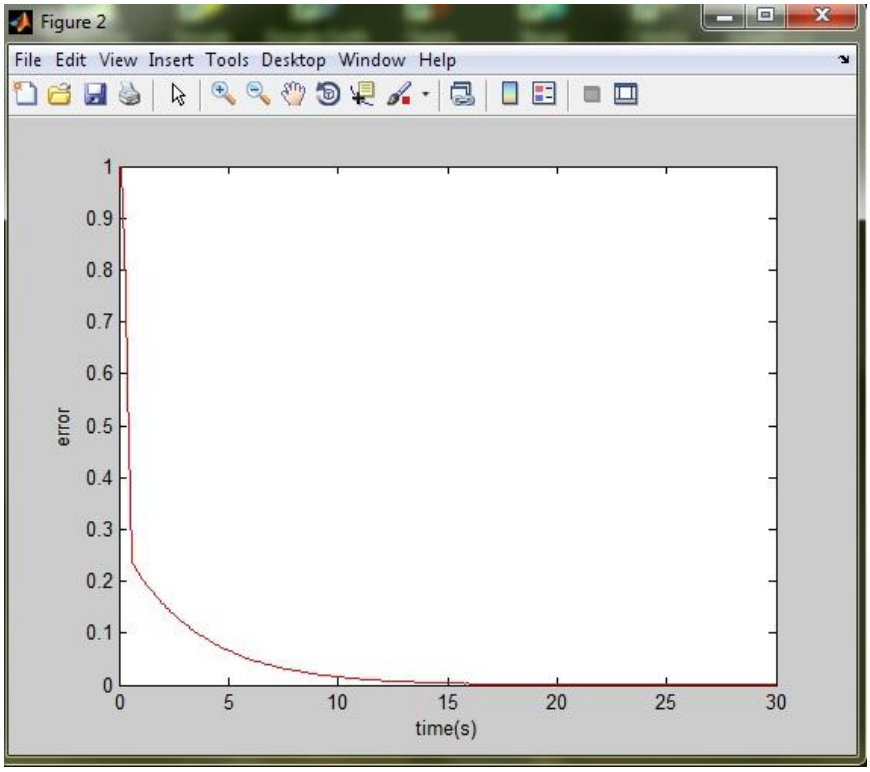
(b) Optimized value of K_i



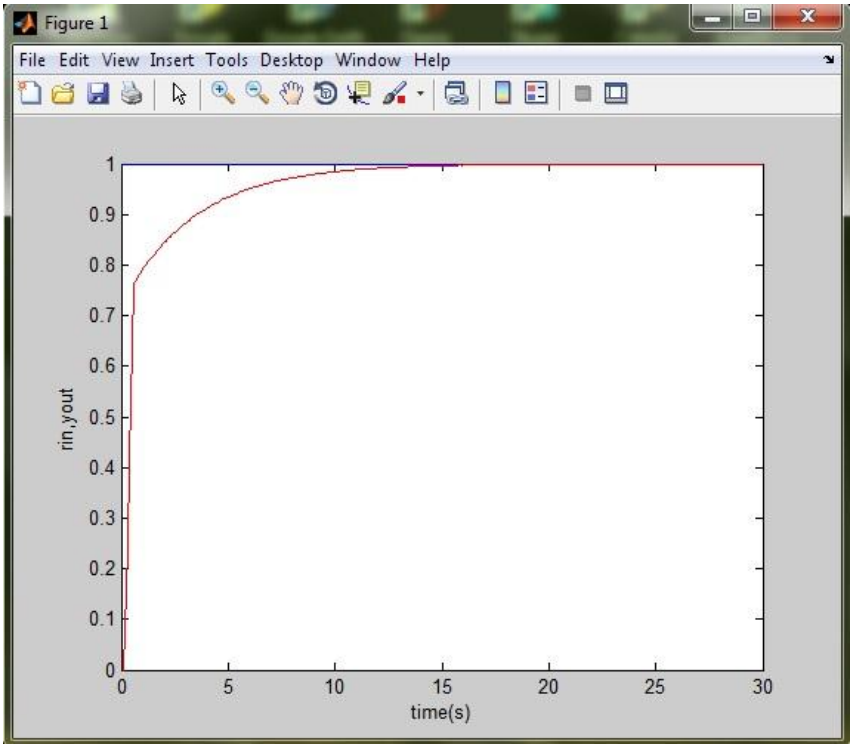
(c) Optimized value of K_i



(d) Manipulated variable u(t)



(e) Error function e(t)



(f) Output response c(t)

Fig 2.6 Matlab simulation result for the system for a step input response

As can be seen from the simulation results, the output response of the system to a step input has a lower rise time and lower overshoot than the output response of the conventioanl PID controller. It has better dynamic properties and steady-state properties.

2.3 Advantages of fuzzy self-tuning PID controller

The followings are the advantages of fuzzy self-tuning PID controller over the conventional PID controller :

i) The traditional PID controller cannot self-tune parameters Kp, Ki and Kd while operating.

ii) Combining fuzzy inference with traditional PID method, self-tuning of PID parameters can be realized.

iii) By designing a fuzzy self-tuning PID controller based on conventional PID, the decisions can be made through fuzzy reasoning rules according to the size, the direction and the changing tendency of the system error together with the dynamic changing of process characteristics.

iv) In practical applications, to different extent, most of the industrial processes exist to be nonlinear, the variability of parameters and the uncertainty of model are very high, thus using conventional PID control the precise control of the process cannot be achieved.

v) The dependence of fuzzy control on the mathematical model is weak, so it isn't necessary to establish the precise mathematical model of the process, and the fuzzy control has a good robustness and adaptability.

vi) The simulation results shows that: compared with the traditional PID controller, fuzzy self-tuning PID controller has a better dynamic response curve, shorter response time, small overshoot, high steady precision, good static and dynamic performance.

2.4 Shortcomings of fuzzy self-tuning PID controller

Although fuzzy self-tuning PID controller gives a better output response than the conventional PID controller, it also has some problems with its design and tuning methods. These problems are mainly due to the vagueness associated with the fuzzy method.

Followings are the disadvatages of a PID controller based on fuzzy logic method :

i) It uses the mode of approximate reasoning and decisions are made on vague and incomplete information similar to that of human beings.
ii) The choice of overall control structure can also be a big problem in some cases.
iii) In designing of the fuzzy logic controller not only the structural parameters need to be designed but also the gain of the conventional controller need to be tuned.
iv) Because of its complicated cross-effects analytical tuning algorithm for these parameters are really difficult.

CHAPTER 3

GENETIC ALGORITHM BASED PID CONTROLLER

3.1 Genetic algorithm based PID controller

3.1.1 Preliminary and background of Genetic Algorithm

Genetic algorithm(GA) uses the principles of evolution, natural selection and genetics from natural biological systems in a computer algorithm to simulate evolution.Essentially, the genetic algorithm is an optomization technique that performs a parallel, stochastic, but directed search to evolve the fittest population.

The idea, in all the system based on Genetic algorithm, was to evolve a population of candidate solutions to a given problem, using operators inspired by natural genetic variation and natural selection.

Biological evolution is an appealing source of inspiration for addressing optimization problems. Evolution is, in effect, a method of searching among an enormous number of possibilities for "solutions." In biology the enormous set of possibilities is the set of possible genetic sequences, and the desired "solutions" are highly favourable organisms—organisms, which are able to survive and reproduce in their environments. Evolution can also be seen as a method for designing innovative solutions to complex problems. The fitness criteria continually change as creatures evolve, so evolution is searching a constantly changing set of possibilities. Searching for solutions in the face of changing conditions is precisely what is required for adaptive computer programs. Furthermore, evolution is a massively parallel search method rather than a work on one species at a time. Evolution tests and changes millions of species in parallel. Finally, viewed from a high level, the "rules" of evolution are remarkably simple: species evolve by means of random variation (via mutation, recombination, and other operators), followed by natural selection in which the fittest tend to survive over others.

3.2 A GA-PID controller

3.2.1 Principles of GA-PID controller

Genatic algorithm is a robust optimization technique based on natural selection. The basic objective of GA is to optimize fitness function. In genetic algorithms, the term chromosome

typically refers to a candidate solution to a problem. In this thesis GA works directly on real parameters. Decimal type GA are equivalent to the traditionally used binary-type GA's in optimization[6]. Decimal-type GA's for computer-based numerical simulation lead to high computational efficiency, smaller computer requirements with no reduction of precision and greater freedom in selecting genetic operator. GA has been successfully implemented in the area of industrial electronics, for instance, parameter and system identification, control robotics, pattern recognition,planning and scheduling.For its use in control engineering, GA can be applied to a number of control methodologies for the improvement of the overall system performance.

3.2.2 Structure and design of GA-PID controller

The structure of a control system with GA-PID as a controller is shown in the figure below. It consists of a conventional PID controller with its parameter optimized by genetic algorithm. The initial population of size N is generated randomly to start the optimization process. The next generation can be obtained through the genetic operators. The genetic opearators are the most important features of GA and are described below.

Genetic operator

The decision to make during implentation of genetic algorithm is the choice of genetic operators that are to be used. The basic genetic operators are;

Reproduction :- By using the values of the performance fitness functions,select the best N/2 induviduals of the current generation to be the parents for producing the next generation.This means that only genetically good individuals are selected to become parents.

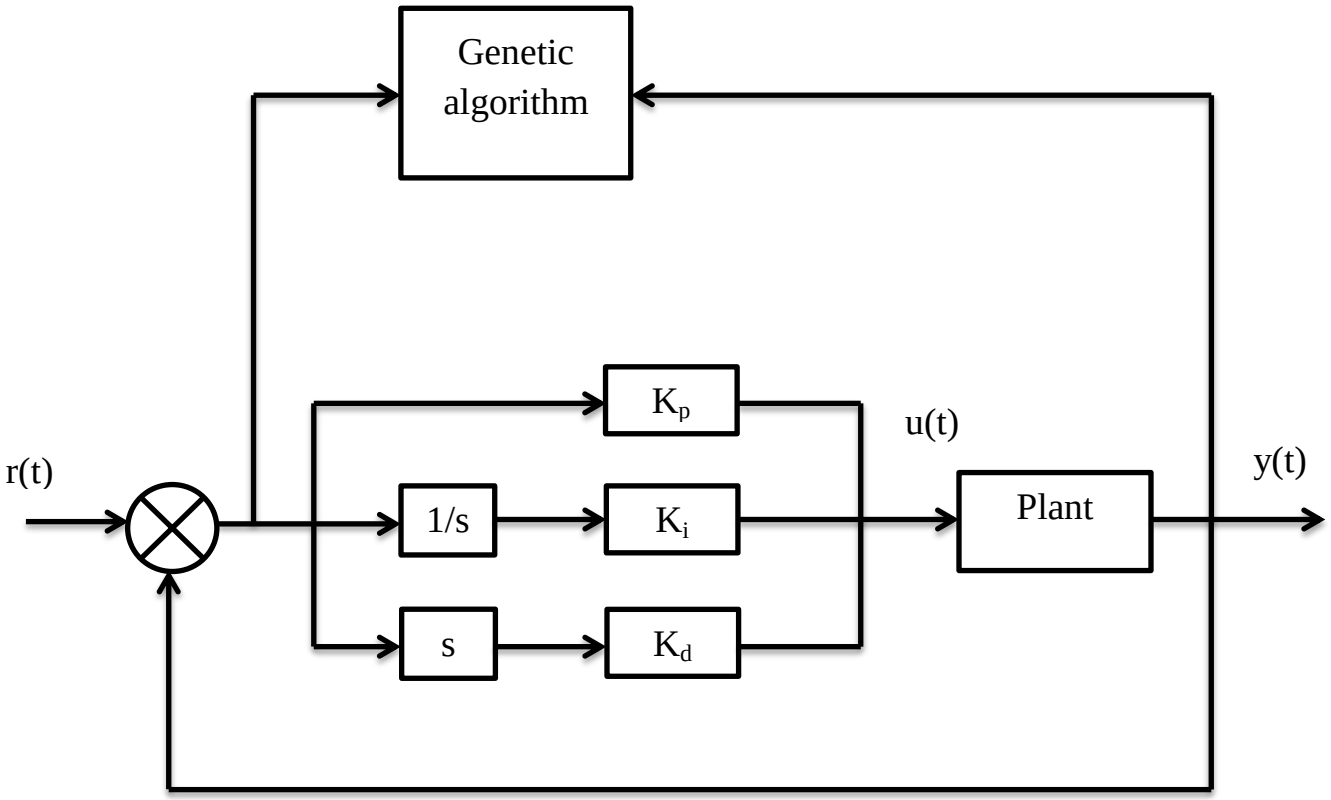


Fig. 3.1 Structure of GA-PID controller

Crossover :-Two parents are randomly selected to exchange the genetic information with each other and two new individuals are generated so as to keep the population size at constant value N.

Cross over opeartion[7] can be mathematically desribed as follows :

If parents are (w_{n1}, k_{f1}) and (w_{n2}, k_{f2}), then

Child-1: $w_n = r * w_{n1} + (1-r) * w_{n2}$

$k_f = r * k_{f1} + (1-r) * k_{f2}$

Child-2: $w_n = (1-r) * w_{n1} + r * w_{n2}$

$k_f = (1-r) * k_{f1} + r * k_{f2}$,where $r \in (0,1)$ is a random number ……..3.1

Here the crossover operator works with real decimal pairs instead of any coded strings.

Mutation :- Mutation plays a secondary role in genetic algorithms. It is needed because, occasionally, chromosomes may lose some potentially useful genetic material. Mutation takes place with a certain probability; thus genetic content of a particular individual gets changed and a new generation is produced. Mutation is important in nature as it brings a change in genetic content of the invdividuals in order to enable them to adapt to a different environment. In the same way, in artificial systems the mutation will direct the search algorithm to a new search space so that a global minima can be found. In our simulations, mutation rate is set to be 0.1.

After mutation,we get a modified mating pool M(k). To form the next generation for the population,we let

$$P(k+1)=M(k) \qquad \text{.....3.2}$$

Where M(k) is the one that was formed by selection and modified by crossover and mutation. Then the above steps repeat, successive generations are produced, and the evolution is modelled.

Search space and fitness landscape

The idea of searching among a collection of candidate solutions for a desired solution is so common that it has been given its own name: searching in a "search space." Here the term "search space" refers to some collection of candidate solutions to a problem and some notion of "distance" between candidate solutions.

Fitness function

A fitness function takes a chromosome as an input and returns a number that is a measure of the chromosome's performance on the problem to be solved. Fitness function plays the same role in GA as the environment plays in natural evolution. The interaction of an individual with its environment provides a measure of fitness to reproduce. Similarly the interaction of a chromosome with a fitness function provides a measure of fitness that the GA uses while carrying out reproduction. Genetic algorithm is a maximization routine; the fitness function must be a non-negative figure of merit.

In this particular situation our main aim is to minimise error and reduce the rise time and overshoot. Hence the fitness function, in this case, is a function of error and rise time.

$$J = \int_0^{\infty} (w_1|e(t)| + w_2u^2(t))\, dt + w_3t_r \qquad \ldots\ldots 3.3$$

Where w_1, w_2, w_3, w_4 are the weight cofficients.U(t) is the output of the controller.e (t) is the error.

The square term of control output is added to overcome the large energy of the controller.

In this paper fitness function is chosen as,

$$f = \frac{1}{J + 10^{-8}} \qquad \ldots\ldots 3.4$$

The term 10^{-8} is added in the denominator of fitness function to avoid it from becoming zero.

Fitness proportionate selection with "Roulette Wheel"

The original GA used fitness proportionate selection, in which the "expected value" of an individual (i.e., the expected number of times an individual will be selected to reproduce) is that individual's fitness divided by the average fitness of the population. The most common method for implementing this, is "roulette wheel" sampling[9]. Each individual is assigned a slice of a circular "roulette wheel", the size of the slice being proportional to the individual's fitness. The wheel is spun N number of times, where N is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation.

This method can be implemented as follows:

1. The total expected value of individuals in the population are summed and the sum is called as T.
2. The above step is repeated N number of times.

A random integer r between 0 and T is choosen. Looping is carried out through the individuals in the population, summing the expected values, until the sum is greater than or equal to r. The individual whose expected value puts the sum over this limit is the one selected.

This stochastic method statistically results in the expected number of offspring for each individual. However, with the relatively small populations typically used in GAs, the actual

number of offspring allocated to an individual is often far from its expected value (an extremely unlikely series of spins of the roulette wheel could even allocate all offspring to the worst individual in the population).

Formalization of Genetic algorithm (GA)

We started with a random population of binary strings of length L.

1. Fitness f(x) of each string x in the population was calculated.
2. We chose (with replacement) two parents from the current population with probability proportional to each string's relative fitness in the population.
3. Cross over was carried out between the two parents (at a single randomly chosen point) with probability P_c to form two offspring. (If no crossover occurs, the offspring are exact copies of the parents.) One of the offspring was selected at random and the other was discarded.
4. We mutated the selected offspring with probability pm and place it in the new population.
5. Step 2 is repeated until a new population is complete.
6. The above process was repeated again from step 1.

Terminal conditions

While biological evolutionary process continues, perhaps indefinitely we would like to terminate our artificial one and find the following:

1.The popualtion string, say ø(k), best maximizes the fitness function. To determine this ,we also need to know the generation number k where the fittest individual existed (it is not necessarily in the last generation). A computer code, implementing genetic algorithm, keeps track of the highest J value, and the generation number and the individual that achieved this value of J.

2.The value of the fitness function J(ø).

The flowchart of simulation for GA-PID controller[4] is as follows:

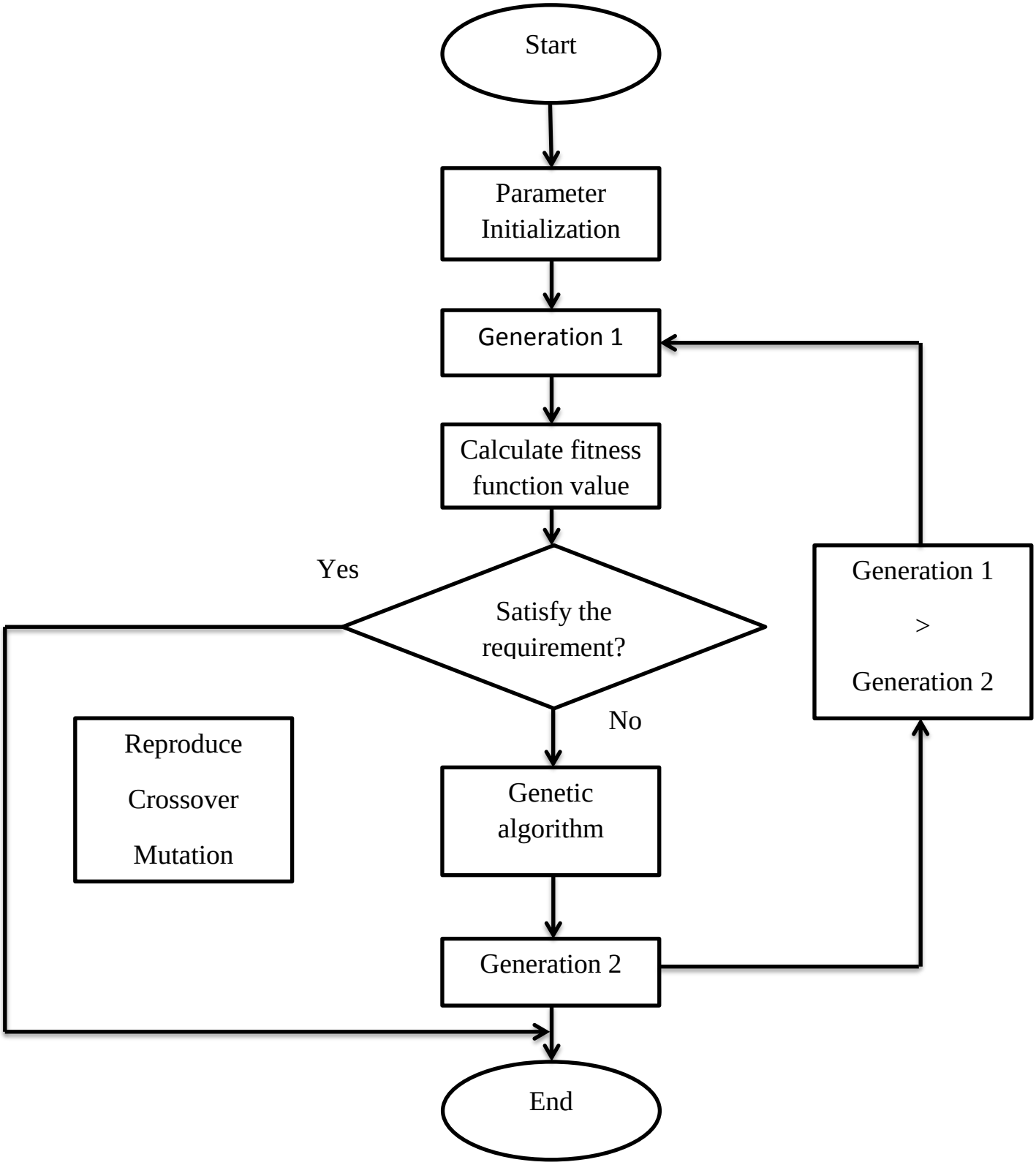


Fig. 3.2 Simluation flowchart for auto-tuning GA-PID controller

3.2.3 Simulation results

Simulation results are obtained using MATLAB for the same transfer function which was used in the cases of conventional PID and fuzzy self-tuning PID controller.

$$G(s) = \frac{500}{s^3+30s^2+1000s} \qquad \text{.....3.5}$$

The size of population of GA is often chosen between [20,100]. For our simulation, we chose the size of population as 40. The number of generation is often chosen between [100,500]. For our case, we chose number of generatons equal to 300. The mutation rate is chosen to be 0.05. The weight co-efficients w_1, w_2 and w_3 are 0.988, 0.001 and 3.0 respectively.The parameter ranges of GA-PID controller are $K_p \epsilon [0,20]$, $K_i \epsilon [0,1]$ and $K_d \epsilon [0,5]$.

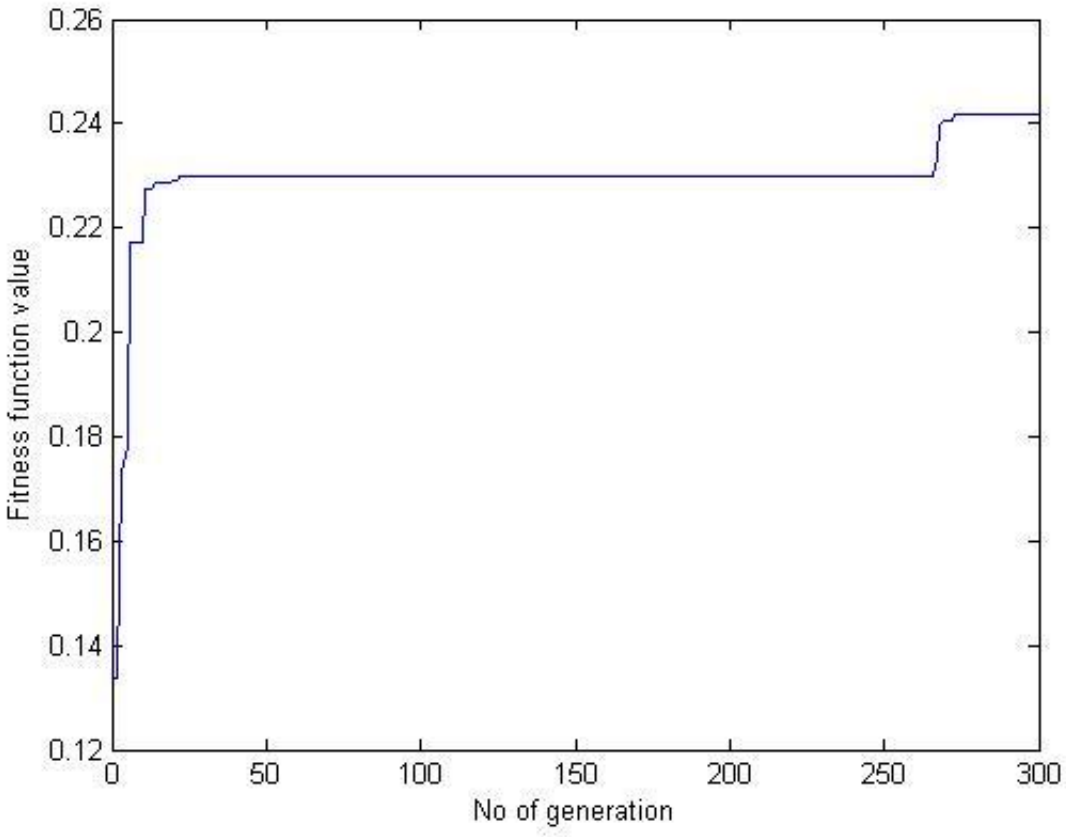


Fig. 3.3 Fitness function plot

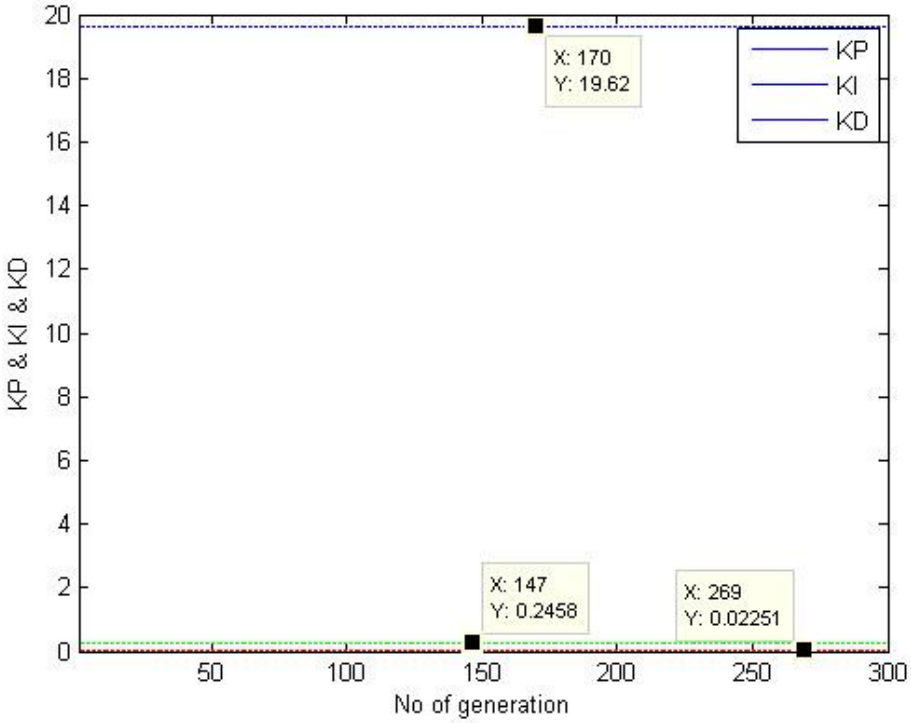


Fig. 3.4 PID parameters (Optimized values)

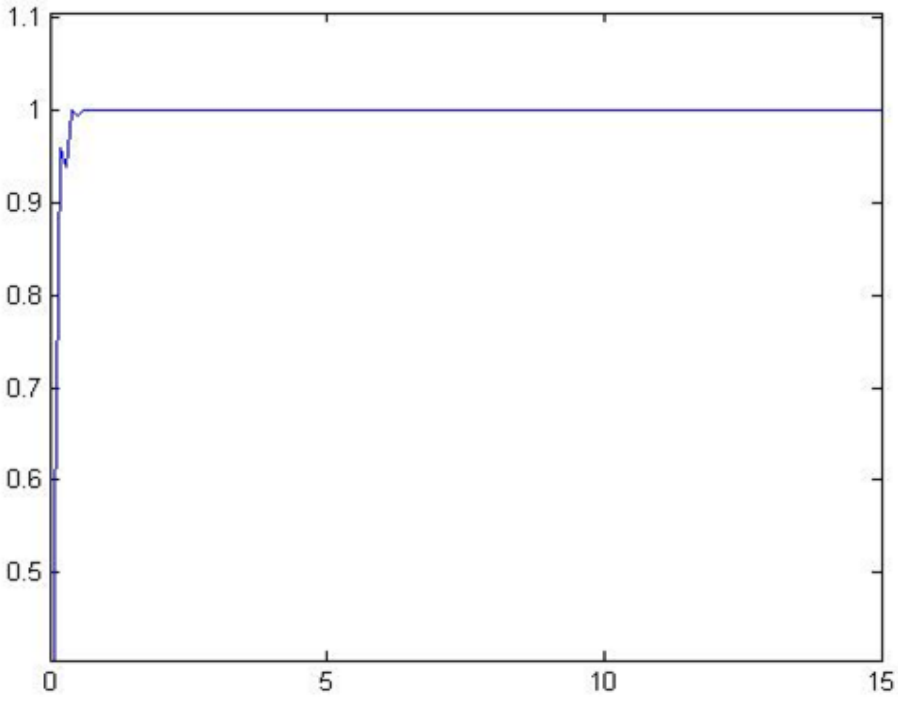


Fig. 3.5 Output response of the GA-PID controller

3.2.4 Advantages of GA-PID controller

i) It is a simple algorithm that is easily understood and implemented.
ii) The algorithm is robust.
iii) GA is a non-linear process that could be applied to most industrial processes with good results.
iv) GA searches a population of points instead of a single solution.
v) GA does not need information about the system except for the fitness function.

3.2.5 Shortcomings of GA-PID controller

For a GA-PID controller, it cannot be guaranteed that the result obtained through the process is the most optimized values although it's near optimum. As GA can different result for each new search for the same system under same conditions. In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem.Therefore, the result may not be the perfectly optmized one.

CHAPTER 4

CONCLUSION

4.1 Conclusion

In this project, we studied the design and tuning methods for PID controller using fuzzy logic and Genetic algorithm. A third-order plant was taken as the control object. Simulation was carried out using MATLAB to get the output response of the system to a step input.The simulation results and the characteristics of both the methods were observed and compared with that of coventional PID controller.

According to the profiling results, the use of above soft-computing techniques resulted in an outputs better dynamic and static characteristics.The response of the system was also faster than in the case of conventional PID controller.The amount of overshoot for the output response was successfully decreased using the above techniques. The application of fuzzy logic to the PID controller imparted it's the ability to tune itself while operating on-line. Similarly, Genetic algorithm enabled the PID controller to get an output which is robust and has faster response.

4.2 Future scope

Future scope of this project involves combination of fuzzy logic and Genetic algorithm for tuning of PID controller. In this method, GA and fuzzy logic account for estimation of gain parameters and ranking basement of GA respectively.

Neuro-fuzzy PID controller can be designed by the implementation of neural network to fuzzy PID controller.

References

[1] J.-S.R. Jang, C.-T. Sun, E. Mizutani. *Neuro-fuzzy and soft computing.*

[2] S. N. Sivanandam, S. Sumathi , S. N. Deepa. *Introduction to fuzzy logic using Matlab.*

[3] Wang-Xiao Kan, Sun Zhong-Liang, Wnglei, Feng Dong-qing. "*Design and research based on fuzzy self-tuning PID using Matlab*". Proceedings of International Conference on Advanced Computing theory and Engineering (2008).

[4] Liu Fan, Er Meng Joo. "*Design for Auto-tuning PID Controller Based on Genetic Algorithms*". IEEE Conference on Industrial Electronics and Applications (ICIEA 2009)

[5] B. Nagaraj, S. Subha, B.Rampriya. "*Tuning Algorithms for PID Controller Using Soft Computing Techniques*".

[6] Melanie Mitchell."*An introduction to genetic algorithms*". MIT Press.

[7] S.S. Ge,member,IEEE, T.H. Lee,Member,IEEE and G.Zhu . "*Genetic algorithm tuning of Lyapunov-based controllers:An Application to a Single-Link flexible Robot System*". IEEE transaction on industrial electronics, VOL. 43,no.5, october 1996.

[8] J. G. Ziegler, N. B. Nichols, "*Optimum setting forautomatic controllers*", Trans. ASME, Vol. 64, pp. 759-768, 1942.

[9] Gopal M. "*Digital control and state variable methods*".